

REMARKS

Claims 1-54 remain pending in the application and reconsideration is respectfully requested in light of the following remarks.

Section 102(e) Rejection:

The Office Action rejected claims 1-3, 8, 15, 17, 20-22, 29, 31, 34-36, 42, 44-46, 52 and 54 under 35 U.S.C. § 102(e) as being anticipated by Young (U.S. Patent 6,560,606). Applicants assert that claims 1-3, 8, 15, 17, 20-22, 29, 31, 34-36, 42, 44-46, 52 and 54 are not anticipated by Young for at least the reasons listed below.

Regarding claim 1, Young fails to teach a method comprising configuring preference values for one or more pluggable components on a first device; and distributing the one or more pluggable components to one or more other devices via a network subsequent to said configuring; wherein the one or more pluggable components are executable within the one or more other devices in accordance with the configured preference values to provide services to users of the one or more other devices.

In contrast, Young teaches a metering and processing system for processing metered information that incorporates configurable processing modules and a configuration manager. (Young, Abstract). Young's system includes "a mechanism for converting the metered information into session data, a processing unit for processing the session data, and a configuration manager." According to Young, the processing unit includes "an execution management framework, and a plurality of plug-ins for processing the session data as directed by the framework with each performing a sub-part of the calculations." According to Young, the plug-in modules reside on each processing unit and a configuration manager "generates a configuration file reflecting user selections of configuration parameters for plug-in execution." (Young, column 3, lines 5-14). Young further teaches that the "configuration manager 150 generates a configuration file for each stage of the pipeline, preferably specifying the configuration data in XML format",

that configuration files are “sent to each stage configuration module” and also that they are “sent to the execution management framework.” (Young, column 10, lines 4 – 14).

Thus, Young’s system does not include *distributing* plug-in modules to the individual machines *subsequent to configuration* of preference values for the plug-in modules on another device. Further, Young describes how “[a] stage configuration module 416 receives configuration files 418 from the configuration manager 150, which define state operations as well as operation of the plug-ins.” Thus, the configuration manager sends configuration information to each stage of the pipeline, but the plug-ins themselves already reside on each processing unit. The actual plug-in modules are not sent by the configuration manager, but rather reside on each individual processing unit of a distributed pipeline.

In response to the above arguments, the Examiner argues that Young’s configuration files are pluggable components that are “executed within each stage to provide configuration and layout for the plug-ins and various other components” (See, Response to Arguments, section A). However, Young’s configuration files are XML documents containing the preference values for his plug-ins. The configuration files, which the Examiner equates to pluggable components, *are not executable within the other devices*. Specifically, Young teaches that the configuration files specify configuration data in XML format (Young, column 10, lines 4-7). As is well known in the art, XML is not an executable language or format. Instead, as Young describes, XML is “a standard, data structuring and formatting language” (Young, column 5, lines 19-20). Contrary to the Examiner’s assertion, Young’s configuration files are clearly not executable. No one of ordinary skill in the art would consider Young’s configuration files to be executable plug-in modules.

Furthermore, even if Young’s executable files were executable, which Applicants maintain they are not, they would not be executable in accordance with the configured preference values *to provide services to users* of the one or more other devices, as recited in claim 1. Young teaches that the configuration files contain configuration parameters

for each stage configuration module and that they are used to configure the stages and plug-ins of Young's pipeline at three levels (Young, column 10, lines 15-56). Young does not describe the configuration as providing services to users of the other devices. Instead, Young describes how other modules, such as the configuration manger, stage configuration manager, and execution management framework, access the configuration files in order to determine the proper functioning of Young's pipeline (Young, column 9, lines 16-33, and column 13, lines 17-29).

Additionally, following the Examiner's line of reasoning, in order to anticipate claim 1, Young would have to teach configuring preference values for his configuration files, which Young does not do. Claim 1 recites, in part, configuring preference values *for one or more pluggable components*. Thus, using the Examiner's interpretation of Young's configuration files as pluggable components, to anticipate claim 1, Young must teach configuring preference values for his configuration files. Young does not do this. Instead, Young teaches using configuration files to store and communicate preferences for the stages and plug-in modules of his pipeline. Young also teaches that his configuration manager "generates a configuration file reflecting user selections of configuration parameters for plug-in execution" (Young, column 3, lines 12-14). Young does not teach that his configuration manager configures parameters for his configuration files. Thus, the Examiner's interpretation of Young's configuration files as executable modules is an incorrect interpretation of Young's teachings.

In light of the above remarks, Applicants assert that the rejection of claim 1 is not supported by the cited art and withdrawal of the rejection is respectfully requested. Similar remarks as discussed above in regard to claim 1 apply to claims 21, 35 and 45.

Regarding claim 15, contrary to the Examiner's assertion, Young does not teach wherein each of the pluggable components comprises a preferences file comprising the preference values associated with the pluggable component. In contrast, Young teaches the use of a single configuration file that is sent to every stage in a distributed pipeline, and that the configuration file contains the configuration information *for all plug-ins*

executing on that device. According to Young, “[t]he configuration file is sent to each stage configuration module for configuring the respective stage.” (Young, column 10, lines 7-9). Young further teaches that a stage includes an input queue, an output queue, and a multithreading process space and that “[t]he process space processes a number of plug-ins ... under the control of an execution management framework.” Furthermore, Young does not teach or suggest that a plug-in includes a preferences file that comprises its preference values. Thus, rather than *each pluggable component comprising a preference file* comprising the preference values associated with the pluggable component, Young teaches that a single configuration file includes the configuration information for a stage, which may include multiple plug-ins.

In response to the above arguments, the Examiner maintains his interpretation of Young’s configuration files as pluggable components. However, as discussed above for claim 1, such an interpretation of Young is clearly incorrect.

Additionally, the Examiner argues that each of Young’s configuration files “is composed of a file containing preferences associated with specific parameters within the configuration file necessary to provide configuration and layout to the stages and plug-ins” (Response to Arguments, section B). However, claim 15, recites that each of the one or more pluggable components comprises a preference file comprising the preference values *associated with the pluggable component*. Using the Examiner’s line of reasoning, where a configuration file is a pluggable component, each configuration file would need to include a preference file including preference values associated with the configuration file. However, the Examiner’s own argument is that each configuration file provides configuration and layout for the stages and plug-ins, not for the configuration file itself.

Thus, in light of the above remarks, Applicants assert that the rejection of claim 15 is further unsupported by the cited art and withdrawal of the rejection is respectfully requested. Similar remarks as discussed above in regard to claim 15 apply to claims 29 and 42.

Section 103(a) Rejection:

The Office Action rejected claims 10-12, 25-26, 39-40 and 49-50 under 35 U.S.C. § 103(a) as being unpatentable over Young in view of Semenzato (U.S. Patent 5,903,728).

Regarding claim 10, Young in view of Semenzato fails to teach wherein the one or more pluggable components is a plurality of pluggable components, wherein each of the plurality of pluggable components are copies of a first pluggable component. The Examiner admits that Young fails to teach wherein each of the pluggable components is a copy of the first pluggable component and relies upon Semenzato for this teaching. The Examiner cites column 3, lines 46-54 and column 6, lines 18-42 of Semenzato. However, at the cited portions Semenzato teaches the use of the Unix fork() system call to duplicate a platform process before executing a plug-in module. The Examiner argues that the Unix fork() call taught by Semenzato can be used to duplicate the configuration files of Young. However, as described above regarding the § 102 rejection of claim 1, the configuration files of Young are XML documents, not executable files. The fork() system call cannot be used to duplicate the configuration files of Young. Additionally, even if the configuration files of Young were to be executable, it is unclear how Semenzato's use of the fork() system call could be combined with Young's pipeline configuration system. For example, Young teaches that the configuration files are distributed to each pipeline server via HTTP (Young, column 10, lines 11-14). Once a configuration file is on a pipeline server, there is no need to then duplicate it using the fork() system call (if the configuration files were executable, which they are not). Additionally, it makes no sense to use the fork() system call to duplicate a process before distribution, as fork() only duplicates an executing process in system memory and such a duplicate is not appropriate for distribution via HTTP as taught by Young.

The Examiner's interpretation of Young's configuration files as pluggable modules, is clearly not compatible with duplication using the fork() system call of

Semenzato. Thus, any combination of Young and Semenzato would fail to result in a system that includes wherein each of the plurality of pluggable components are copies of a first pluggable component. The removal of the § 103 rejection of claim 10 is therefore respectfully requested. Similar arguments as those above regarding claim 10 apply to claims 25, 39 and 49, as well.

Regarding claim 12, Young in view of Semenzato fails to teach sending each of the plurality of pluggable components to the corresponding one of the plurality of devices via the network. The Examiner cites portions of Young that describe distributing configuration files via HTTP (Young, fig 3, column 8, lines 48-58, column 10, lines 9-20, column 13, lines 32-42 and column 17, lines 16-21). However, claim 12 also requires that each of the plurality of pluggable components be a copy of a first pluggable component (via claim 10). Semenzato teaches the use of a fork() system call to duplicate executing processes and indeed the Examiner relies upon this teaching in the rejection of claim 10. However, as described above regarding claim 10, the fork() system call is incompatible with the XML based configuration files of Young. Furthermore, since the fork() system call duplicates processes executing in system memory, the resulting duplicates cannot be distributed via HTTP as suggested by the Examiner. Thus, the combination of Young and Semenzato fails to teach sending each of the plurality of pluggable components to the corresponding one of the plurality of devices via the network. Similar arguments also apply to claims 26, 40, and 50.

The Office Action rejected claims 13-14, 27-28, 41 and 51 under 35 U.S.C. § 103(a) as being unpatentable over Young in view of Davis et al. (U.S. Patent 5,742,829) (hereinafter "Davis").

In regards to claim 13, Young in view of Davis fails to teach executing the batch file on the first device. The Examiner cites portions of Davis (column 11, lines 50-67, column 12, lines 61 - column 13, line 38) that describe using a SMSLS batch file to configure a client computer from a server computer. However, Davis teaches that "although the SMSLS batch file ... [is] located on the client server, the processing (or

execution) of these files occurs on the processor of the client computer” (Davis, column 11, lines 60-64). Thus, Davis actually teaches away from executing the batch file on the first device, which corresponds to the server machine in Davis’ system. When modified in view of the cited teaches of Davis, Young’s system would include executing a batch file on each of Young’s pipeline servers. Therefore, the Examiner’s proposed combination of Young and Davis files to teach executing the batch file on the first device.

Furthermore, the combination of Young and Davis further fails to teach generating a batch file comprising one or more configuration entries for the one or more pluggable components, wherein each configuration entry includes information specifying one of the one or more pluggable components; information specifying one of the preference values for the specified pluggable component; and a new value for the specified preference values, as recited in claim 13. In contrast, Davis teaches that his batch file invokes a client setup executable that copies various files of the client operating system onto the client computer. Davis does not mention including, in his batch file, configuration entries for pluggable components. Davis also fails to teach that such a configuration entry includes information specifying a pluggable component, information specifying a preference values for the specified pluggable component, and a new value for the specified preference value. As Young fails to teach the any use of a batch file, as admitted by the Examiner, Young fails to correct any of the deficiencies of Davis’ teachings.

Thus, the Examiner’s proposed combination of Young in view of Davis fails to result in a system that teaches the limitations of claim 13 as described above. Similar arguments as applies to claim 13 apply to claims 27, 41 and 51.

The Office Action rejected claim 4 under 35 U.S.C. § 103(a) as being unpatentable over Young in view of Hammond (U.S. Patent 6,637,020), claims 5-6, 23, 37 and 47 as being unpatentable over Young in view of Barrett et al. (U.S. Patent 6,611,876) (hereinafter “Barrett”), claim 7 as being unpatentable over Young and Barrett, as applied to claim 5 above, and in view of Hammond, claims 9, 24, 38 and 48 as being

unpatentable over Young in view of Foltan et al. (U.S. Patent 6,667,972) (hereinafter “Foltan”), claims 16, 18, 30 and 32 as being unpatentable over Young in view of Lawrence (U.S. Patent 6,629,113), claims 19, 33, 43 and 53 as being unpatentable over Young in view of Muschett et al. (U.S. Patent 6,026,437) (hereinafter “Muschett”). Applicants traverse each of these rejections for at least the reasons given above in regard to the independent claims.

Furthermore, in regard to both the section 102 and section 103 rejections, Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

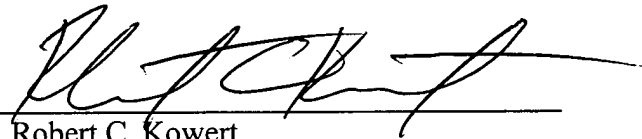
Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-46501/RCK.

Also enclosed herewith are the following items:

- ☒ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Fee Authorization Form authorizing a deposit account debit in the amount of \$
for fees ().
- ☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: December 15, 2004